*Лицей Академии Яндекса*

*Детский Технопарк Кванториум*

Проект PyQT

***Редактор дипломов***

Выполнил:

Подвысоцкий К. Д.,

ученик 11 класса

Руководитель проекта:

Замятина О. В,

преподователь Лицея Академии Яндекса

Калининград 2024

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc183180812)

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ 4](#_Toc183180813)

# ВВЕДЕНИЕ

На одном из уроков в Яндекс Лицее, на которых мы обсуждали тему для проектов, к нам пришли из администрации «Кванториума» с прозьбой создать приложение, которое создавла бы серию дипломов использую инофрмацию из Excel-таблицы. Эта тема мне показалось интересной, ведь эта проблема действительно может беспокоить многих, именно поэтому я её и выбрал.

Проблема – отсутствие автоматизированного процесса создания однотипных «дипломов». («диплом» в моем проекте – это все, что связано с бумажной наградой: дипломы, грамоты, сертификаты и т.п.)

Цель – создать приложение, которое позволит автоматизировано создавать серию «димломов» по заданному шаблону.

Задачи:

1. Продумать, какие функции должен выполнять мой проект.
2. Продумать схему приложения и пользовательский путь.
3. Реализовать приложение исползуя язык Python и библиотеку PyQt6
4. Сделать презентацию и пояснительную записку
5. Защитить проект

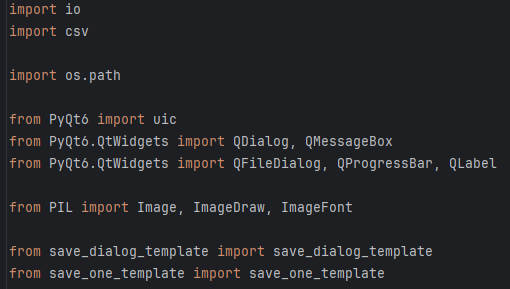
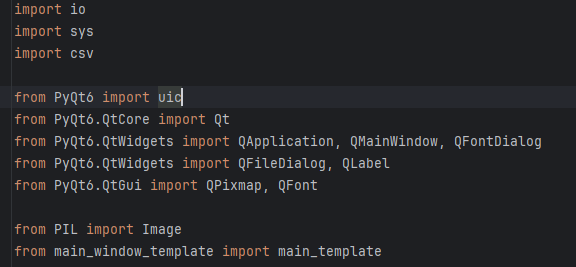
# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Приложение должно выполнять функции:

* Принимать на вход Excel-таблицу с базой данных и пустой шаблон «диплома».
* Извлекать названия колонок – будущие «виджеты» («виджетами в моём проекте назвается текст с именем колонки из загруженной таблицы, на место которого будет загружены данные из соответствующей колонки»).
* Размещать «виджеты» на шаблон и редактировать их стиль.
* Добавлять дополнительную информацию на шаблон (текст, картинки).
* Предпросмотр на отдельном примере из таблицы.
* Сохранять серию готовых «дипломов» в выбранную папку.

# СХЕМА ПРОЕКТА

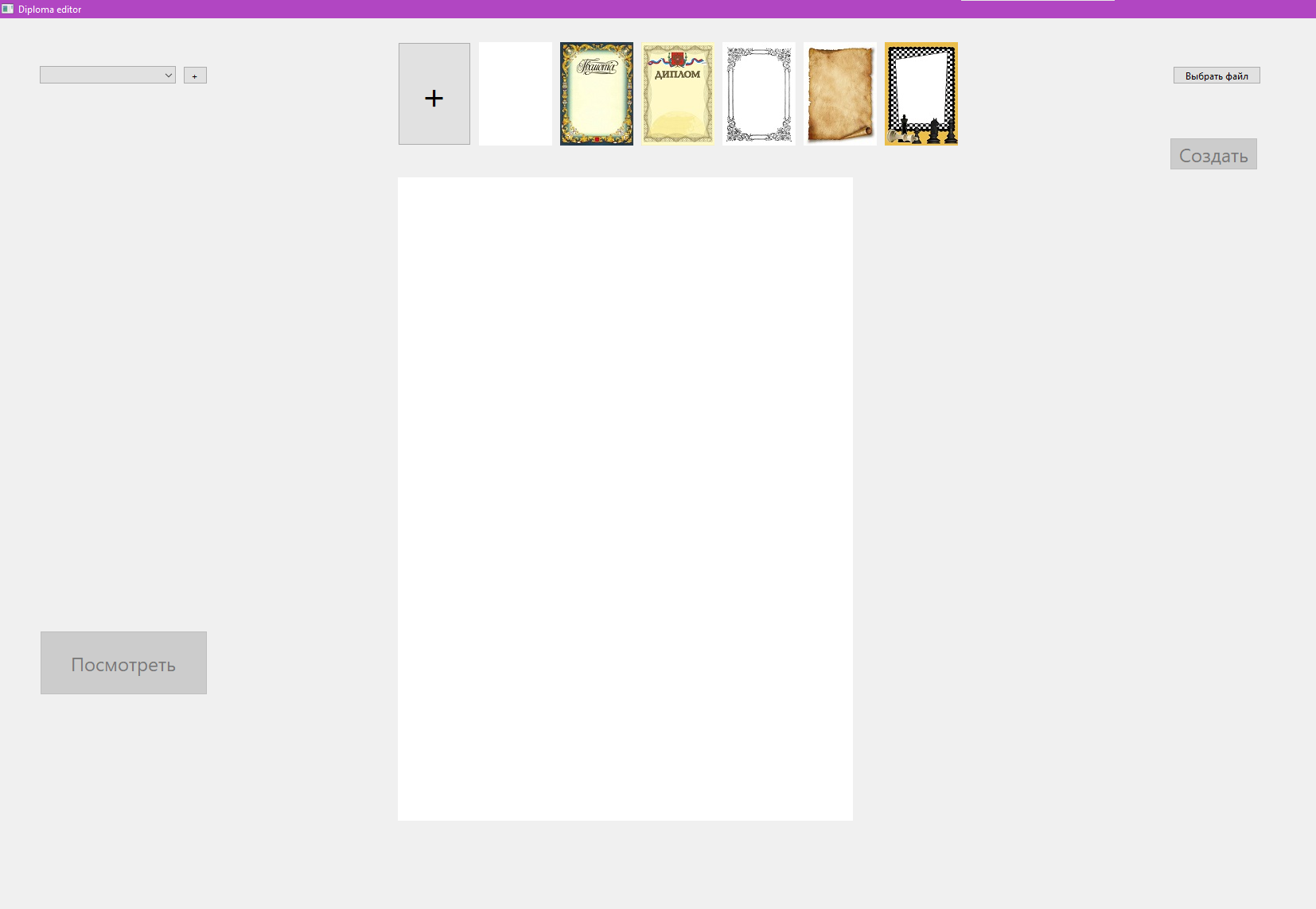
# ИСПОЛЬЗУЕМЫЕ БИБЛИОТЕКИ

* **PyQt6** – создание приложения и дизайна
* **pillow** – создание картинок
* **openpyxl** – открытие excel файлов
* **sqlite3** – работа с SQL
* **csv** – работа с csv файлами
* **auto-py-to-exe** – создание exe файла

# РЕАЛИЗАЦИЯ ПРОЕКТА

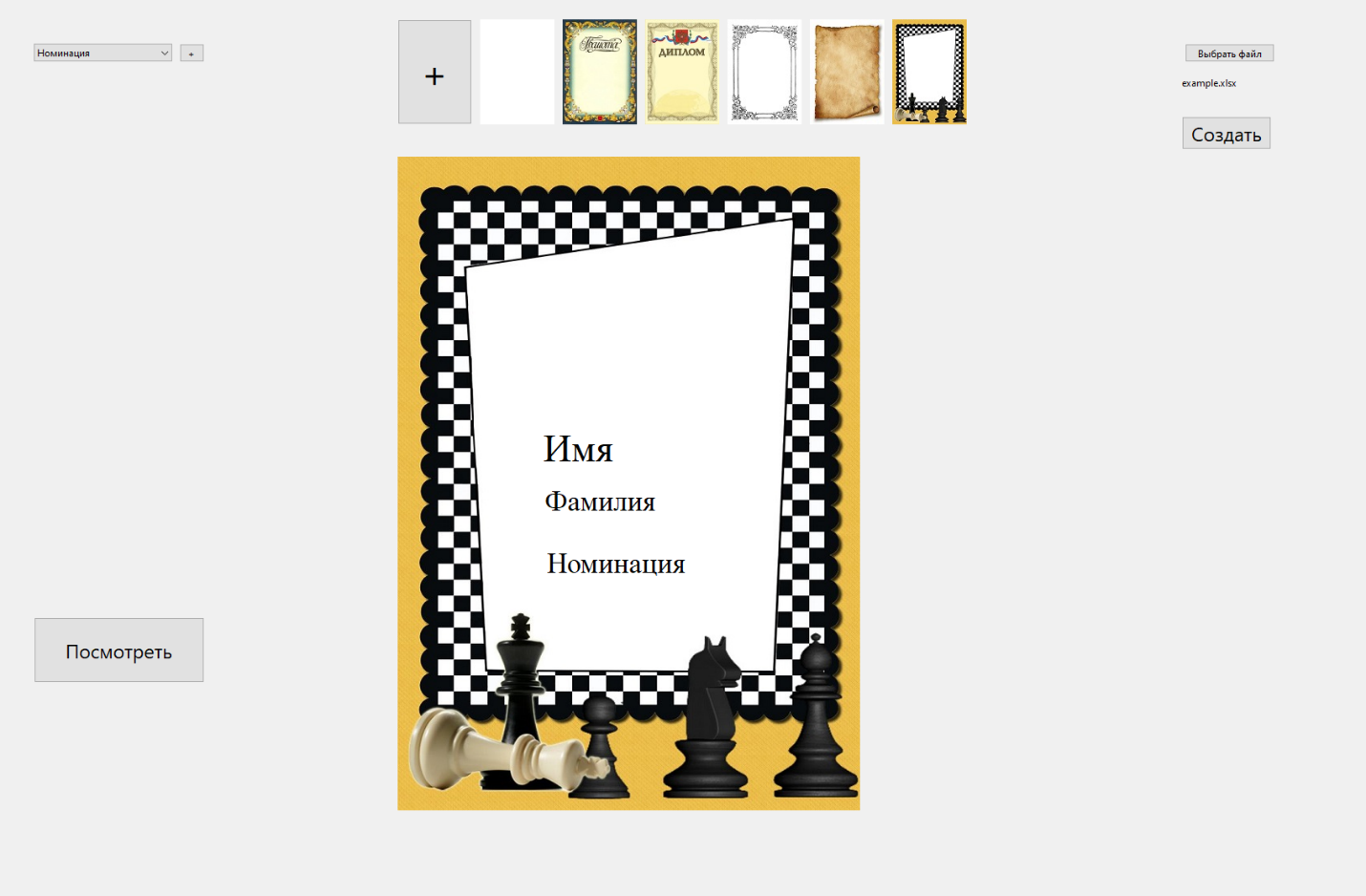
Реализацию проекта я начал с создания ui-шаблона главного окна в QtDesigner. Я добавил:

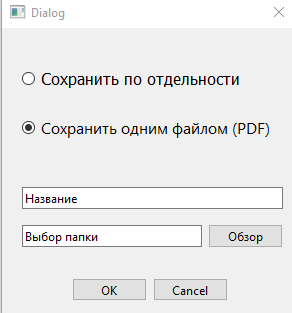
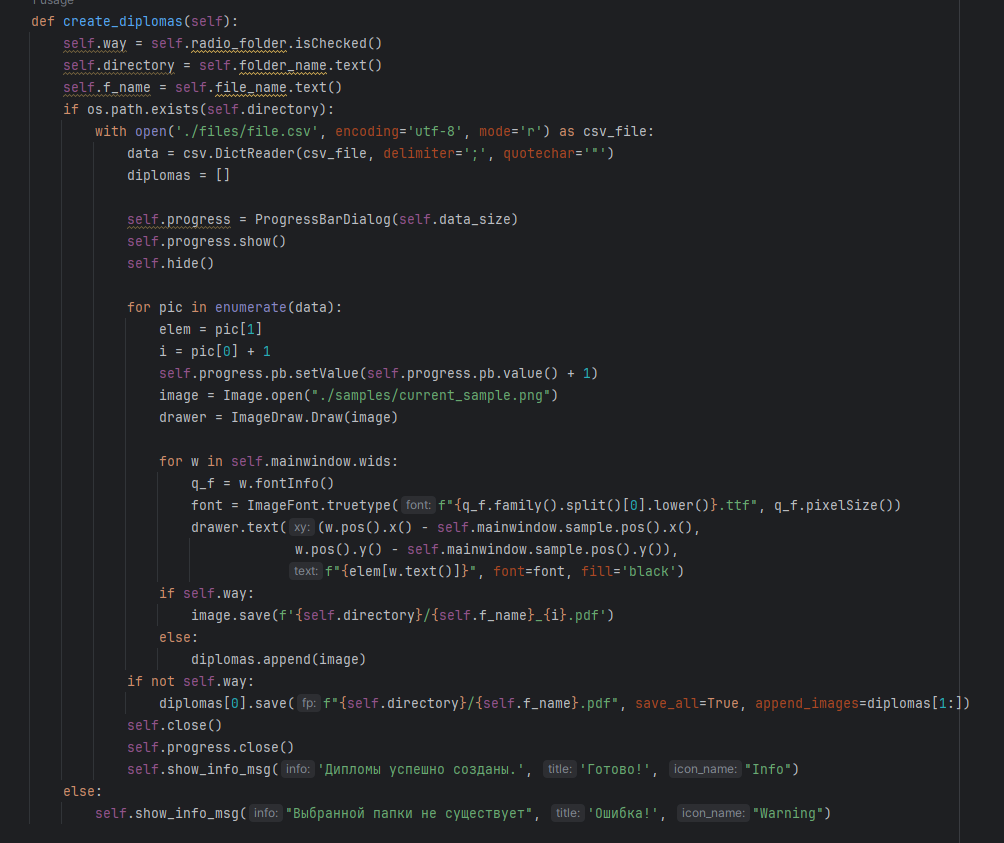
* Кнопку QPushButton импорта Excel файла и QLabel, в которую указывается название таблицы.
* Кнопку QPushButton импорта своего пустого шаблона.
* QLabel, в которую будет импортироваться шаблон.
* QComboBox, в который будут загружаться заголовки колонок таблицы и QPushButton, которая будет добавлять «виджет» на шаблон.
* QPushButton, которая будет открывать окно предпросмотра.
* QPushButton, которая будет открывать диалог сохранения «дипломов».



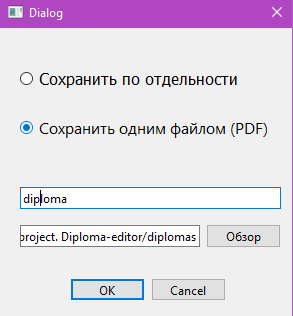
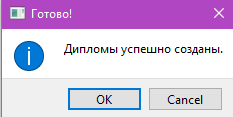
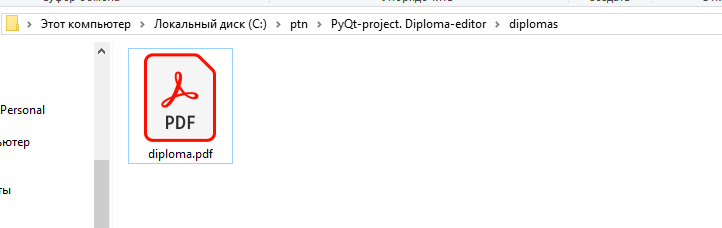
После импортирования ui в код, я начал реализацию импортрования Excel таблицы. Для этого я написал функцию, с помощью библиотеки openpyxl, xlsx2csv(), которая с помощью QFileDialog принимает xlsx файл и форматирует таблицу в csv. С csv файлами в моем проекте работать удобнее, чем с файлами, которая предоставляет openpyxl. Назваие таблицы выводится в QLabel под кнопкой. Затем, функция file\_input(), передаёт названния колонок в QComboBox.

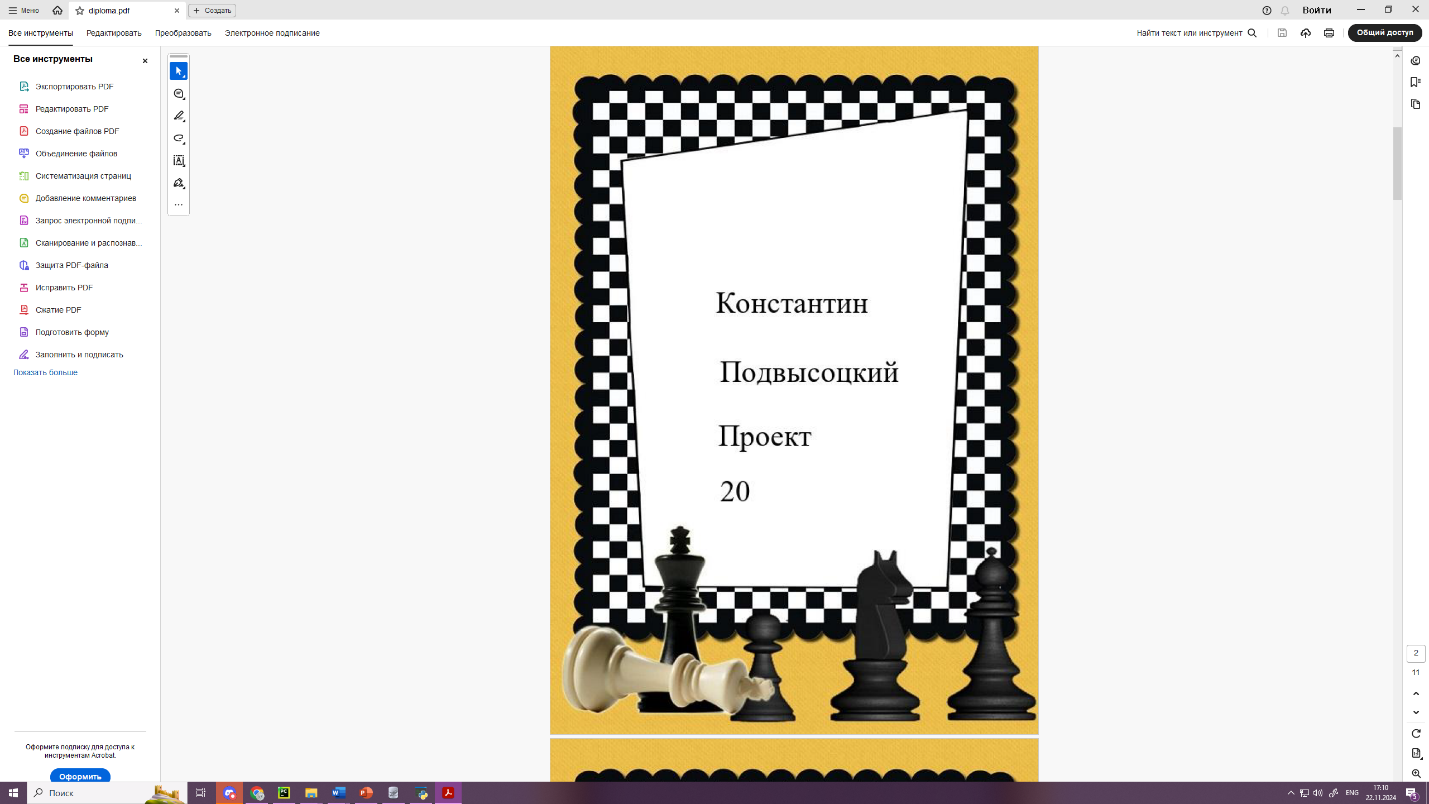
Далее я преступил к создании функции выбора шаблона add\_sample(name), которая принмает имя name картинки с пустым шаблоном и размещает его в в QLabel с помощью QPixMap. Изначально выбирается белый шаблон. Также можно с помощью кнопки QPushButton, которая импортирует шаблон и с помощью функции add\_sample\_dialog(), которая вызывает QFileDialog, разместить шаблон. Либо с помощью курсора и функции mousePressEvent можно нажать на размещенные сверху QLabel с QPixMap иконками, и выберутся готовые шаблоны. Реализация нажатия происходит засчет сравнивания координат курсора с координатам иконок.

Далее я реализовал добавление иконок. Оно происходит при нажатии на QPushButton, и добавляется «виджет» QLabel, на кнопку F на клавиатуре можно изменить ему шрифт и размер с помощью QFontDialog. С помощью мышки и MauseEvent его можно брать и двигать. При нажати на Delete, он удаляется. Эти «виджеты» размещаются на шаблон.

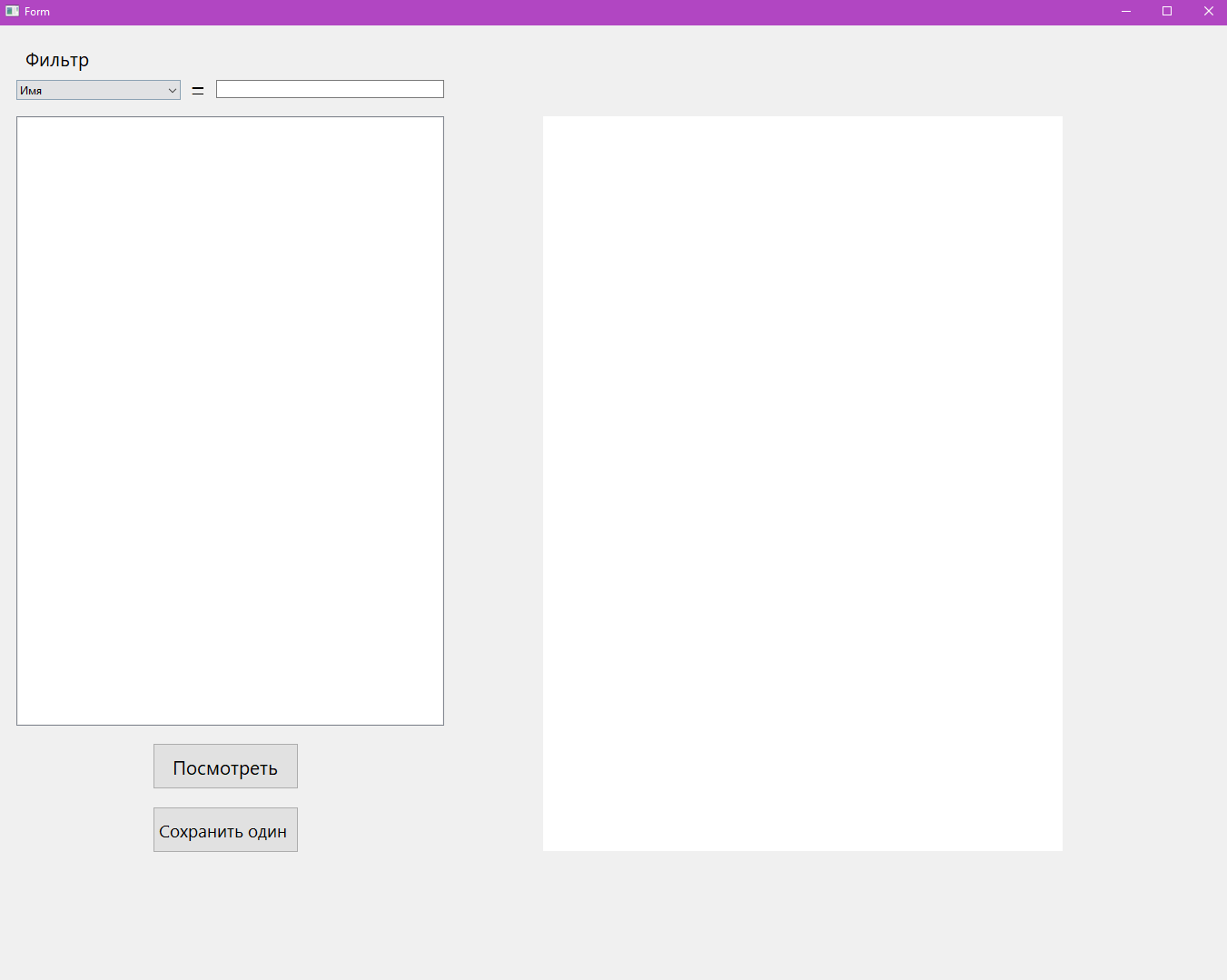
* Далее я приступил к реализации создания дипломов. При надатии на QPushButton открывается диалоговое окно. Его я также создал с помощью QtDesigner.
* Я добавил:
* QRadioButton, который определяет способ сохранения файлов: одним PDF файлом или несколькими по отдельности каждый «диплом».
* QLineEdit, который принимает название
* QLineEdit, который принимает название папки и QPushButton, который вызывает QFileDialog. getExistingDirectory(), который вводит в QLineEdit путь до папки, куда будут сохранятся файлы.
* QAcceptButtonBox, который имеет две кнопки: ok и cancel.

При нажатии на кнопку ok программа сначала проверяет существование папкки. Проверку на существование папки я взял из Интернета, она реализуется с помощью os.path. Если папка не существует, то всплывает QMassageBox с ошибкой. Если папка существует, то начнется процесс сохранения с помощью функции create\_diplomas(). Функция перебирает все «виджеты» на шаблоне и извлекает соответствующите позиции из csv-таблицы. Далее она определяет координаты «виджета» на шаблоне (вычитает из координат «виджета» соответствующие координаты шаблона в окне) и на пустой шаблон подставляет соответствующую информацию с помощью библиотеки Pillow и drawer.text. В зависимости от выбранного QRadioButton, файлы сохранятся в выбранную па пку выбранным способом. Во время загрузки высплывает диалог с QProgressBar, к оторый отображает процент загрузки и по завершении высплывет QMassageBox, информируя о том, что

все файлы создались.

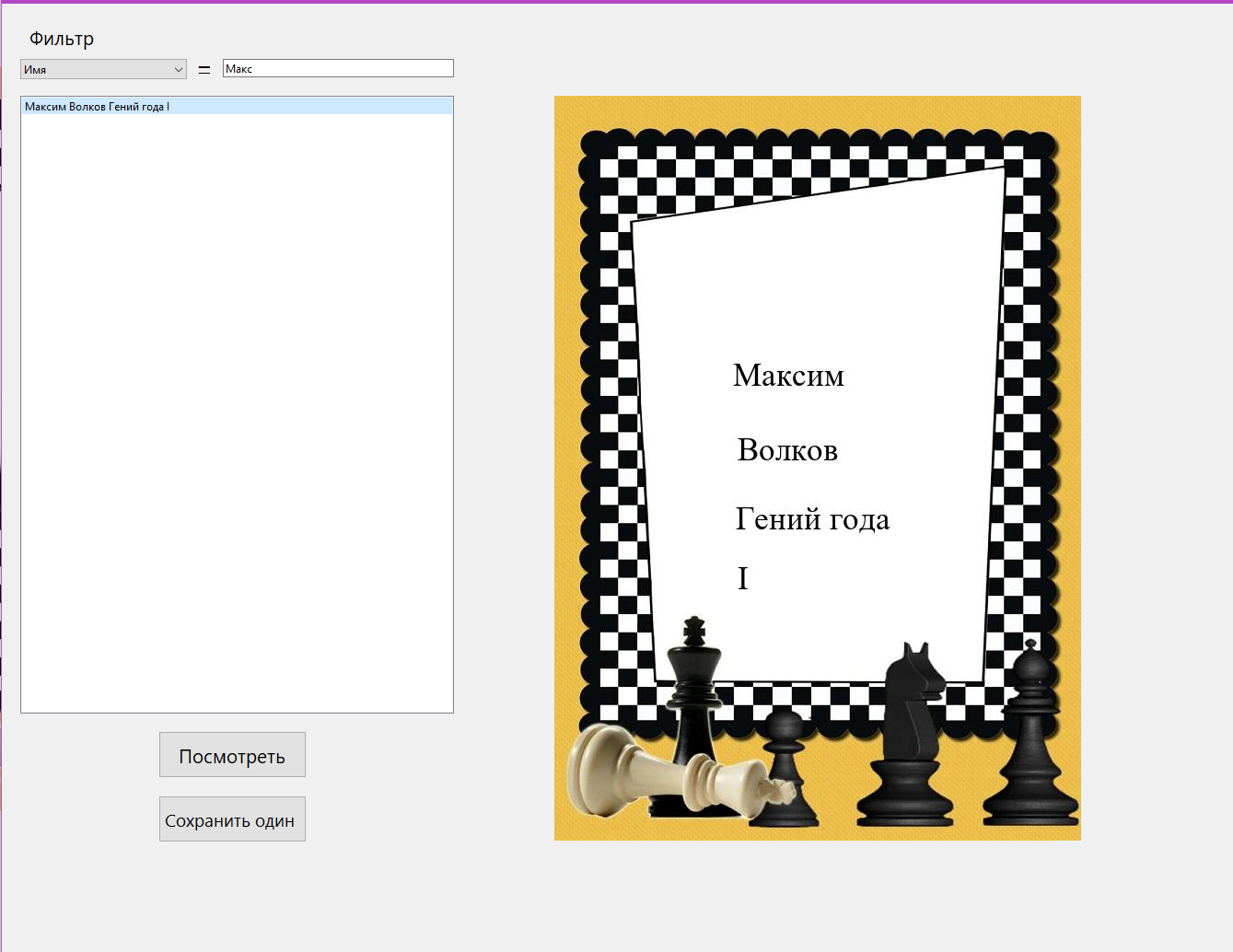


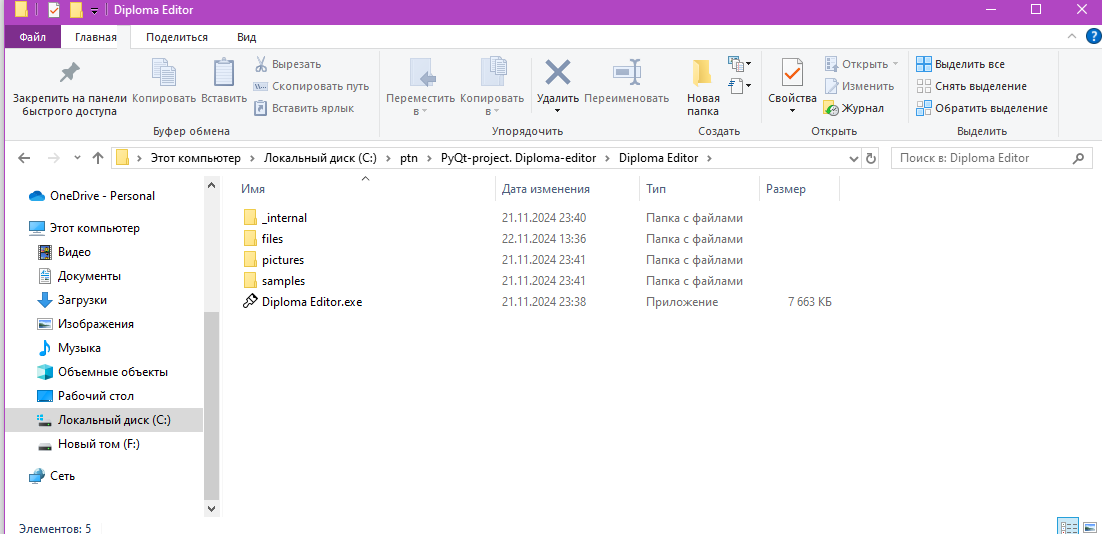
Далее я принялся создавать окно предпросмотра в QtDesigner.



Я добавил:

* QComboBox, который имеет те же названия колонок.
* QLineEdit, который принмает запрос пользователя.
* QListWidget, куда будут выводится результаты запроса пользователя.
* QLabel, куда аналогично главному окну вставляется QPixMap с шаблоном.
* QPushButton «Посомтреть», которая подтверждает запрос.
* QPushButton «Сохранить один», которая сохраняет 1 выбранный файл аналогично тому, как сохранялись все файлы.

Сначала я вернулся на главное окно и добавил, что при загрузке Excel в csv, с помощью функции csv2sql() создаётся SQL база данных. Окно предпросмотра вызывается с помощью QPushButton «Посмотреть» на главном окне, Пользователь выбирает в QComboBox имя колонки из таблицы и вводит в QLineEdit запрос. Все, что похоже на запрос с помощью SQL-запроса SELECT LIKE выбереться и появится в QListWidget. При нажатии на любой элемент списка, белый шаблон заменется на готовый диплом для выбранного элмента. Если пользователя что-то не устроит, то он может вернуться на главное окно и подпраить. Или он может сохранить один «диплом» для выбранного элемента аналогично тому, как сохранялись все «дипломы».

Далее я принялся с помощью модуля auto-py-to-exe создавать exe файл своего проекта, что далось мне с большим трудом. Я добавил иконку ввиде кисточки и название “Diploma Editor” – «редактор дипломов».

На этом моя работа над приложением пока что завершилась.